Public-Key Management System using Hamming Distance for Mobile Ad Hoc Network

Seok-Lae Lee[†]*, Joo-Seok Song*[††]

[†]***KISA(Korea Information Security Agency) 78 Garak-Dong, Songpa-Gu, Seoul, 138-803, Korea***
[††]***Yonsei University, 134 Shinchon-Dong, Seodaemoon-Gu, Seoul, 120-749, Korea***

Summary
Mobile ad-hoc networks have various implementation constraints such as infrastructure-free, non-trusted authority, node mobility, and the limited power and small memory of mobile device. And just like wired networks, various security issues such as authentication, confidentiality, integrity, non-repudiation, access control, availability, and so on, have been arisen in the mobile ad-hoc networks. In this paper, we focus on the authentication of these security issues because it is quietly affected by the characteristics of networks. We propose the public-key management system based on the concept of Hamming distance for mobile ad-hoc network. Particularly, we propose an authentication protocol that improves the certificate repository size of each node as $\log_2 N$ and assures to make a trusted certification path from one node to another, using the concept of Hamming distance.

Key words:
Public-key, Private key, Certificate, Ad hoc network, Hamming Distance.

1. Introduction

Mobile ad-hoc networks [1] provide convenient infra-free communications over wireless channels. And a small size networking device in these networks communicates with other devices using the radio frequency by peer-to-peer network model. Recently, wireless ad-hoc networking technology has been applied to various applications such as military tactical networks, personal area networks, sensor networks, collaborative networking, and disaster area networks [2][3]. But it is difficult to implement mobile ad-hoc networks due to physical constraints such as non-trusted authority, node mobility, and the limited power range and small memory of mobile device, and due to the security issues such as authentication, confidentiality, integrity, non-repudiation, access control, availability, and so on. Recently, there are many efforts to resolve the characteristics of mobile ad-hoc networks with various security issues [4][5][6][7][10][11].

In [4][5][6], Luo *et al* distributes the functions of certification authority through a threshold secret sharing method [8][9] and scalable multi-signature mechanism, in which each node holds a secret sharing key and multi-nodes in a local neighborhood jointly provide complete services. In [10][11], Capkun *et al* proposed a fully self-organized public-key management system that doesn't need any trusted authorities or centralized servers in mobile ad-hoc network and doesn't assign specific missions to a subset of nodes. This system allows users to generate their public-private key pairs, to issue certificates, and to perform authentication regardless of the network partitions and without any centralized services. Furthermore, the authors proposed the simple repository construction algorithm for public-key authentication and showed that two users in a mobile ad-hoc network can perform key authentication based on their local information, even if security is performed in a self-organized way. Last, the authors bring up the problems that minimize the size of the certificate repositories.

So we focus on the problem to minimize the size of the certificate repository. In our paper, we propose an authentication protocol that makes the size of the certificate repository as $\log_2 N$ using the concept of Hamming distance [12], with assuring to make trusted certification paths [13] from one node to another. We present the public-key management mechanism to issue, update and revoke certificates between nodes based on Hamming distance. And we propose a certification path construction algorithm and analyze the performance of the algorithm to prove enable to construct a certification path, even if the repository size is limited as $\log_2 N$. Furthermore, our protocol can construct a certification path for nodes regardless of decreasing or increasing the number of nodes in mobile ad-hoc network.

This paper is organized as follows: In Chapter 2, we explain theorem, definitions, and concepts for introducing Hamming distance into our authentication protocol. In

Manuscript received August 5, 2006.
Manuscript revised August 25, 2006.

Chapter 3, we explain the basic PKI management based on the concept of Hamming distance. In Chapter 4 and 5, we propose the certification path construction algorithm and analyze the performance of proposed algorithm. And in the last chapter, we conclude and give some remarks on our future work.

2. Theorem, Definitions and Concepts

In our System, user (or node) authentication protocol is to verify the authentication path from a user (Tn) to another user (Vn) as the following way: ① User obtains the certification path from him to Vn using his certificate repository and the certificate repositories of other nodes over the mobile ad-hoc network. We call this process as certification path construction. ② User Tn checks that all certificates on the path are valid (or not revoke) and correct (not false). We call this process as certification path validation. We focus on the certification path construction because we can use the path validation algorithm of RFC3280 for the certification path validation. First, we explain the requirements that need to certify nodes (or users) in mobile ad-hoc network and present our ideas to meet the requirements. Table 1 shows the requirements and ideas.

Table 1: Authentication requirements in mobile ad hoc network

Requirements	Ideas
• All certificates issued to nodes must be distinguished.	• Assign unique ID to all certificates
• All nodes can authenticate without help from trusted authorities or centralized server.	• Offer self-organized authentication method
• All nodes must always provide authentication method for each other regardless of power range of mobile node.	• Construct certificate path between nodes using the concept of Hamming Distance
• It needs to minimize the number of nodes which one node stores and manages.	• Improve the repository size of any node as $\log_2 N$
• It needs to minimize certificate path length from Tn to Vn.	• Future work
• Because random nodes join and leave, it must provide network scalability.	• Provide the issuance, update and revocation of certificate, and change the network size by the number of nodes
• Overall networks should be ensured security and reliability. All networks must offer integrity, authentication, confidentiality and non-repudiation.	• Use the PKI certificate

First, we explain the basic characteristics of Hamming distance between two integers and the concept of Hamming distance for improving the repository size of each node and enabling certification path construction.

Theorem 1: For $a \in Z_n$, define $R_a = \{x \in Z_n \mid HD(a,x) = 1\}$, then $|R_a| = \log_2 N$, where $Z_n = \{0, 1, 2,..., N-1, 2^m, m > 0\}$

> **Proof:** For arbitrary integer $a \in Z_n$, the number of integers which the Hamming distance in comparison with a is "1" is the same as the number of integers which the Hamming weight in comparison with "0" is "1". Generally, the number of integers that the Hamming weight in comparison with "0" is r is computed to ${}_mC_r$. Here, m means $\log_2 N$ as defined in Theorem 1. Thus, for arbitrary number a, the number of integers that Hamming distance is "1" is equivalent to m ($=\log_2 N$) as ${}_mC_1$

And we use the following 4 definitions to simplify the explanation for our protocol.

Definition 1: Certificate ID (Identification Number) within any mobile ad-hoc network is an element of Z_n, $\{0, 1, 2,..., N-1, 2^m, m > 0\}$, and the maximum size of network is defined as N. However, because the nodes within mobile ad-hoc network are changeable, the number of nodes that involved in actual network is defined as Nr.

Definition 2: The IDs of two nodes, a and b, are defined as a_{ID} ($\in Z_N$) and b_{ID} ($\in Z_N$) and the Hamming distance between these IDs is defined as $HD(a, b)$ (or HD)

Definition 3: A node a within mobile ad-hoc network stores and manages some of public-key certificates within network at its own repository in order to construct certification path. At this time, the node a is defined as parent node (PN) and the set of nodes that are stored and managed in the repository of PN is defined as child nodes (CN). Besides, one of CN is defined as CN_i. Each node ID meets the following condition: $HD(PN, CN_i) = 1$.

Definition 4: When node a tries to authenticate node b, a trusted certification path from node a to node b is defined as $Auth_p(a \to b)$. At this point, let's define node a as Tn and node b as Vn. A case for searching trusted path is represented as $Auth_p(Tn \to_{tri} Vn)$ between

Tn and Vn. And if there exists a trusted path, it defines as $Auth_p(Tn \rightarrow_{suc} Vn)$.

In this paper, it assigned arbitrary element within Z_n to certificate ID in order to identify all nodes. At this time, certificate ID is different concept with serial number of public-key certificate x.509 v3 [13] and is managed by using the field of "Subject Name" or "Subject Alternative Name" within certificate. To solve the problem of certification path construction and self-organized certificate management, public-key certificates of all nodes are assigned ID each node and these certificates have particular inter-relationship by using HD in order to construct certification path even though any nodes don't exist. As for this matter, chapter 4 and 5 will address it through certification path construction and simulation. In addition, the number of nodes that a node has to manage by using "Theorem 1" and "Definition 3" in mobile ad-hoc network will become $\log_2 N$. It will explain in the following example as below. According to this concept, the basic operation for issuing, updating and revoking the public-key certificate of each node will be addressed in chapter 3. And the following method will be used to ensure the scalability of mobile ad-hoc network.

- The number of nodes at current network, Nr, is decided by pre-negotiation between nodes, and thereby the size of virtual network (N) is decided. At this point, there is a relation of $N/2 \le Nr < N$ between N and Nr.
- All nodes that exist in mobile ad-hoc network are assigned ID sequentially.
- If the actual number of nodes in the network due to mobility of nodes is less than $N/2$, the maximum size of the network should be changed to $N/2$, At this point, nodes that have IDs more than $N/2$ have to be assigned new IDs less than $N/2$. This process accomplished by certificate updating process.
- If the number of nodes in the network is more than N, nodes are sequentially assigned IDs by the order of ($N, N+1, N+2, \ldots$). And then each node exchanges public-key certificate with the node that Hamming distance is "1". Surely, the size of the network changes to $2N(=2^{m+1})$ from N.

Figure 1 and 2 briefly describe the characteristics explained the above. Figure 1 represents relationship between integers that Hamming distance is "1" about the integer set of $N=8$. Namely, the number of integers that Hamming distance with "001" is "1" is 3 $(=\log_2 8)$ such as {"000", "011", "101"}. It agrees with Theorem 1. At this time, if the ID of PN is "001", the IDs of CN are {"000", "011", "101"} respectively. Namely, R_{001} becomes {"000", "011", "101"}. According to Theorem 1 and Definition 3, PN only manages public-key certificates of $\log_2 N$ nodes at its own repository. Namely, in figure 1, a node "001" only manages the public-key certificates of nodes {"000", "011", "101"}. In figure 2, the certification path from Tn to Vn exists 6 and the shortest paths are ("000" → "010" → "110") and ("000" → "100 → "110"). Our certification path construction algorithm is to find such certification paths.

(Assumption 1) In this paper, on the assumption that all nodes in mobile ad-hoc network are active, we implement the basic operation for issuing, updating and revoking certificate and certification path construction algorithm.

Node ID	000	001	010	011	100	101	110	111
000		●	●		●			
001	●			●		●		
010	●			●			●	
011		●	●					●
100	●					●	●	
101		●			●			●
110			●		●			●
111				●		●	●	

R_{001}

Fig. 1 The set of IDs with HD=1

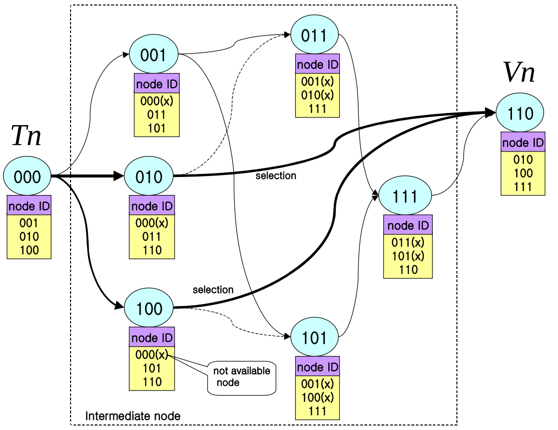


Fig. 2 The concept of Certification Path Construction

3. Basic Operations for Public-key Management

In chapter 2, we explained how to decide nodes that a node manages in ad-hoc network in view of certification path construction. As mentioned at the previous chapter, a node in ad-hoc network implements the generation, update, revocation, management of public-key certificates only belonging to nodes at its own repository. In this chapter, we will explain the issuance, update, and revocation of public-key certificates for nodes based on the concept that explained in the previous chapter.

3.1 Initialization for mobile ad-hoc network

Mobile ad-hoc network is initialized from the following process: ①The network size N is decided through pre-negotiation between nodes involved in mobile ad-hoc network. ②According to N, each node chooses one integer within Z_n. At this time, each node sequentially chooses his ID by referring to its own IP (Internet Protocol) address that was exchanged at pre-negotiation stage. ③After finishing ID assignment to each node, each node exchanges public-key certificate with nodes that have $HD = 1$. And then each node stores and manages the public-key certificates at its own repository.

3.2 Certificate Issuance

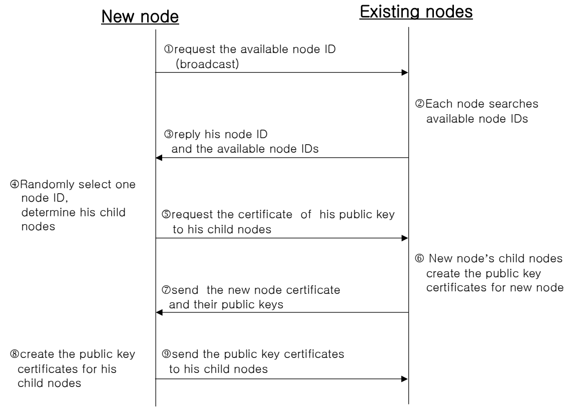

New node
Existing nodes
①request the available node ID (broadcast)
②Each node searches available node IDs
③reply his node ID and the available node IDs
④Randomly select one node ID, determine his child nodes
⑤request the certificate of his public key to his child nodes
⑥ New node's child nodes create the public key certificates for new node
⑦send the new node certificate and their public keys
⑧create the public key certificates for his child nodes
⑨send the public key certificates to his child nodes

Fig. 3 The issuing process of public-key certificate for new node

A new node that wants to participate in mobile ad-hoc network basically has to receive $ID (\in Z_n)$ from existing nodes in the network, and then exchanges certificates with nodes that have $HD = 1$ with its own ID. Next, we will explain how to assign ID and issue certificate to a new node. First of all, new node broadcasts its intention for involving to all nodes in mobile ad-hoc network. Nodes that receive the message reply its own ID and available IDs. If there is no available ID, it sends only its own ID. New node randomly selects one of available IDs as its own ID. Secondly, new node decides the set of nodes that have $HD = 1$ with its own ID. We call it CN (the set of child nodes). And then new node exchanges certificates with CN. In this stage, new node requests CN the certificate for its own public-key and issues certificates for the public-keys of CN using his private key. We call new node PN (Parent Node) from the viewpoint of CN. Figure 3 describes the issuing process of public-key certificate for new node.

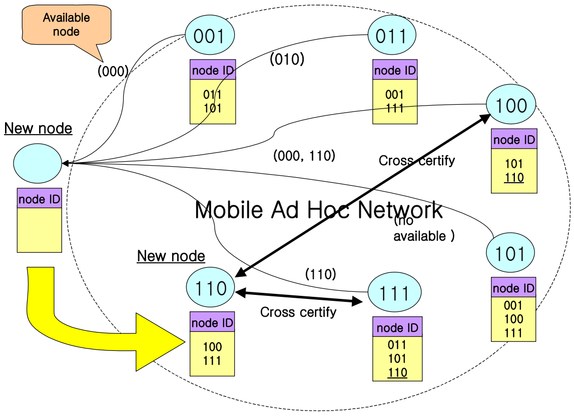


Fig. 4 Certificate Issuance for new node

In Figure 4, it describes that new node obtains available IDs ("000", "010", 110") from nodes in the network and be part of ad-hoc network by choosing its own ID as "110". Also, new node exchanges public-key certificates with nodes ("100", "111") that Hamming distance is "1" with its own ID and then stores them at its own repository. In figure 4, we shows only node ID, because it is important to know the IDs of nodes that a node manages rather than public-key certificate itself.

3.3 Certificate Update

In case of updating public-key certificate in mobile ad-hoc network, there are two cases to be considered. One case is for extending the validity period of public-key certificate and the other is for reducing the scale of mobile ad-hoc network. First, in case of extending the validity period of certificate, the node (PN) that has a certificate that will be expired soon requests CN_i to update their certificates before expiration date. The node (CN_i) that receives this request sends PN 's certificate after updating it and keeps the updated certificate at its own repository. Figure 5 briefly describes certificate update of node "110".

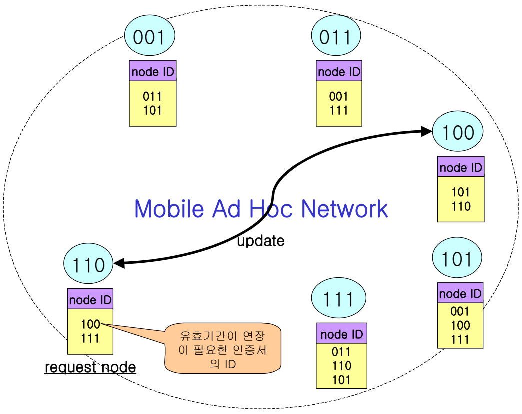


Fig 5 Certificate update: case1

Next case is for reducing the scale of mobile ad-hoc network. This belongs to the case that the size of network is smaller than $N/2$ as conceptually explained in chapter 2. At this time, the target node for updating is the case that the value of node ID is the same or larger than $N/2$. Like Figure 6, if some part of the node is revoked or moved to other networks, the remaining nodes will be "001", "011", and "100". At this time, the size of network is smaller than $N/2$(=4). Therefore, node "100" must update his ID as available node ID("00" or "10") because it is larger than $N/2$(=4). If node "100" is updated as "10", it exchanges certificates with his child node ("11").

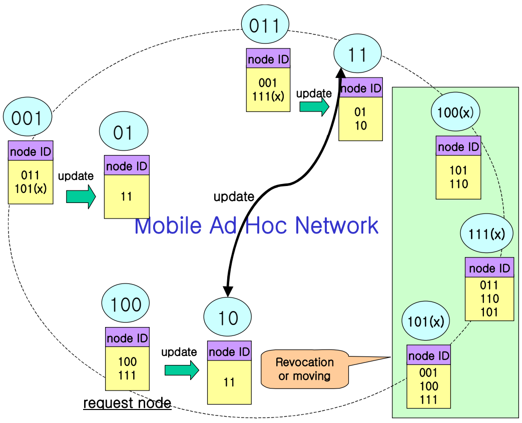


Fig. 5 Certificate update: case2

3.4 Certificate Revocation

While a node (revocation requesting node) detects an illegal activity of any particular node in the network, certificate for the node will be revoked. At this time, a node that detects an illegal activity notifies certificate of the illegal node to all nodes. If a node that receives this notification manages certificate of the illegal node, the node deletes certificate of the illegal certificate from its own repository and then registers CRL (Certificate Revocation List) for implementing revocation process. Afterward, the result of revocation will be notified to revocation requesting node. Figure 7 describes the revocation process.

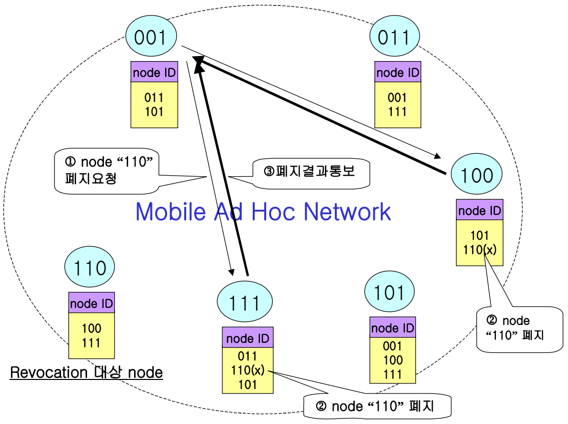


Fig 7 Certificate revocation

4. The Path Construction Algorithm using Hamming Distance

4.1 Certification Path Construction Algorithm

Certification path construction algorithm refers to find one trusted certification path from Tn to Vn. In figure 8, namely, there are six trusted certification paths from Tn to Vn. One of these paths is constructed from Tn to Vn. And then if Vn's certificate is verified that there is no problem, Tn can trust Vn.

This paper presents an algorithm illustrated in figure 9 to search a trusted certification paths from Tn to Vn by using the repository information of each node and the characteristics of Hamming distance explained in chapter 2 in order to construct trusted certification paths. First, the terminology that used in figure 9 will be explained briefly. Path_nodes[] represents the set of certificates on the path from Tn to Vn in the process of certification path construction. The number of real nodes in mobile ad-hoc network is Nr in spite of the network size N, because any nodes can't exit in mobile ad-hoc network. The algorithm proposed in this paper is divided into two parts. One is a part of computing Hamming distance between Vn and the CN_i of PN. The other is a process to replace PN with one node among CN of PN in case that Hamming distance is not "1". If the Hamming distance is "1", the path construction algorithm is successfully terminated. Even if $HD(PN, Vn)$ is not "1", the algorithm can be successfully terminated when there is CN_i $(i = 0, 1, ..., \log_2 N - 1)$ that satisfies $HD(CN_i, Vn) =$ 1. If $HD(CN_i, Vn)$ is not '1', PN is replaced by randomly choosing one of CN. And then by using CN of the replaced PN, it also verifies to satisfy $HD(CN_i, Vn) = 1$. As repeating this computation, a trusted certification path can be found. Sometimes, there is a case that is impossible to construct paths. Besides, even if the size of mobile ad-hoc network is $N(= 2^m)$, the number of real nodes (Nr) that exist in current network can be much smaller than N. In this case, a path can't exist or can't be found. In case not finding a path, if we repeat the path construction algorithm, we may find a certification path.

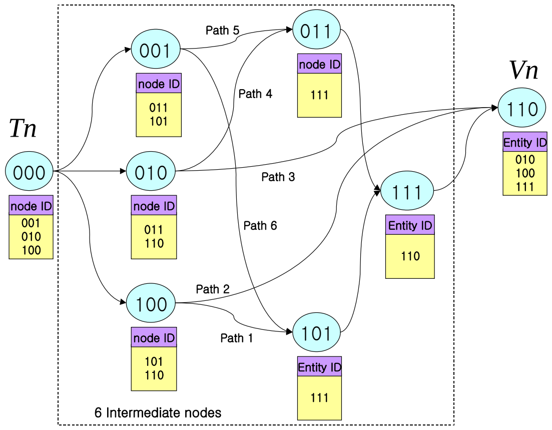


Fig 8 Certification paths between Tn and Vn

In Figure 8, Tn issues certificates to three nodes such as {"001", "010", "100"} and then stores and manages them at its own repository. At this time, Tn becomes PN and {"001", "010", "100"} belong to CN. If Tn doesn't issue certificate to Vn directly, it searches a certification path to Vn using intermediate nodes. At this point, PN randomly chooses one of its own CN for continuing the next process. Namely, one of {"001", "010", "100"} can be chosen as PN. If "010" as PN is chosen, path construction is completed (path: "000" → "010" → "110"). However, if "001" as PN is chosen, path construction is not completed. Therefore, one out of R_{001} ("011" or "101") has to be chosen as PN. At this point, if "011" is chosen as PN, the path will be {"000" → "001" → "011" → "111" → "110"}.

In generally, the path construction algorithm is similar to routing algorithm in the ad-hoc network. In this paper, algorithm performance is defined in the perspective of how to accurately find certification path rather than how to obtain the shortest certification path. By using algorithm performance proposed in [4], it is defined as following.

$$P_b(A_{HD}, r, Z_N) = \frac{\left|\{(Tn,Vn) \in Z_N \times Z_N : Tn \rightarrow_{suc} Vn\}\right|}{\left|\{(Tn,Vn) \in Z_N \times Z_N : Tn \rightarrow_{tri} Vn\}\right|} \quad (1)$$

Here, A_{HD} : The certification path construction algorithm using Hamming distance

r : $(N - N_r)/N$

In chapter 5, we present the simulation result that evaluates algorithm performance.

```
1. initialization
  - $path\_length = 0,\ path\_nodes[0] \leftarrow T_n,\ PN \leftarrow T_n$
2. compute the Hamming distance between $T_n$ and $V_n$
  - if $HD(PN, V_n) = 1$, then go to step5
3. compute then Hamming distance between $CN_i$ and $V_n$
  3-1. i = 0, path_length = path_length + 1
  3-2. randomly select $CN_i$, $CN_i$ ? $R_{PN}$
      (a) if $CN_i$ is not available, then
          - $i \leftarrow i+1$
          - if $i < \log_2 N$, then go to step3-2
          - go to step5(Fail)
      (b) if $HD(CN_i, V_n) = 1$, then go to step5
      (c) if $i < \log_2 N$, then $i \leftarrow i+1$ and go to step3-2
4. randomly select $CN_j$ and replace $PN$ by $CN_j$
  4-1. j = 0
  4-2. randomly select $CN_j$, $CN_j \leftarrow R_{PN}$
      (a) if $CN_j$ has been already processed or not available, then
          - $j \leftarrow j+1$
          - if $j < \log_2 N$, then go to step4-2
          - go to step5(Fail)
      (b) replace $PN$ by $CN_j$
          - $path\_nodes[path\_length] \leftarrow PN$
          - $PN \leftarrow CN_j$
      (c) go to step3-1
5. output($path\_length$, $path\_nodes[]$) and stop
```

Fig 9 Certification path construction algorithm

4.2 Multi-hop Problem

Since the power range of all nodes is limited to one hop, it is required to improve the certification path construction algorithm proposed in the above for constructing the certification path between nodes separated from one hop. In this section, the certification path construction algorithm consists of a certification path tracing algorithm and a certificate acquisition algorithm. A certification path tracing algorithm means that *Tn* finds a path from its own *ID* to *Vn*'s *ID* through the *ID*s of all nodes on mobile ad-hoc network. For this purpose, Each node independently has to manage its own *ID* list of nodes on mobile ad-hoc network. After finishing to find a path by certification path tracing algorithm, *Tn* obtains certificates required to verify *Vn* by practical communication with the nodes on mobile ad-hoc network. This process is defined as a certificate acquisition algorithm. The certification path tracing algorithm will not be addressed here since it is similar with the certification path construction algorithm proposed in the above, but only the certificate acquisition algorithm in multi-hop will be explained.

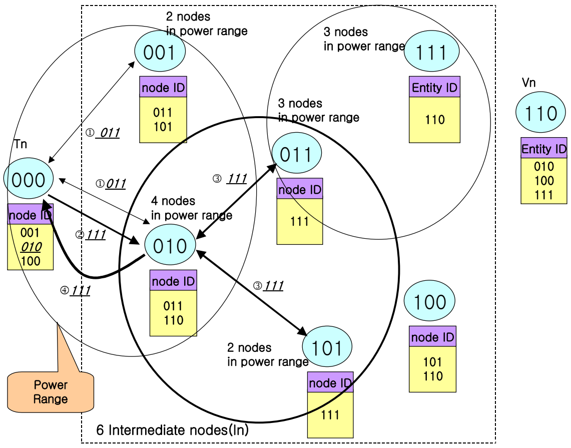


Fig 10 Certificate acquisition method for Multi-hop

The basic method of the certificate acquisition algorithm is to obtain certificates on the path found by the certification path tracing algorithm from the intermediate nodes within power range of *Tn* in advance. If the certificates obtained from this way don't complete a perfect path, *Tn* requires to find missing certificates by choosing one of the intermediate nodes within power range. At this time, the node that connects to the most nodes within power range of the intermediate nodes is chosen as an intermediate node. The certificates on the path are obtained by repeating this step. For example, suppose that the searched path by the certification path tracing algorithm in *Tn* is {"000" → "010" → "011" → "111" → "110"}, the certificates of nodes ("011", "111") should be obtained using intermediate nodes even though the certificates of nodes ("000", "010", "110") of the certificates on the path are already obtained by {*Tn*, certificate stored in the *Tn*'s repository, *Vn*}. Fortunately, the certificate of node "011" is easy to obtain because it is managed in the repository of nodes ("010" and "001") on the power range of *Tn*, but node "111" is out of the power range of *Tn*. Thus, the help from the other node is necessary. In Figure 10, *Tn* requires node "010" to find the certificate of node "111", because node "010" is including the most nodes (4 nodes) within its own power range. The intermediate node "010" confirms that there is the certificate of node "111" within its own power range. Fortunately, since there is the certificate of node "111" that is managed by CN ("011") of node "010", it can be sent to Tn. Through this method, even though the certificates on the certification path is out of the power range, Tn can obtain all certificates on the path for verifying the certificate of Vn.

5. Simulation result for path construction algorithm

The main goal of this simulation is to verify a completion level of our proposed algorithm in figure 9. Thus, this simulation intends to confirm two situations. One is to analyze the length of certification path according to the number of nodes that involved in the mobile ad-hoc network. The other is to analyze the length of certification path and the success probability of path construction in case that the number of nodes on mobile ad-hoc network is smaller than $N(=2^m)$.

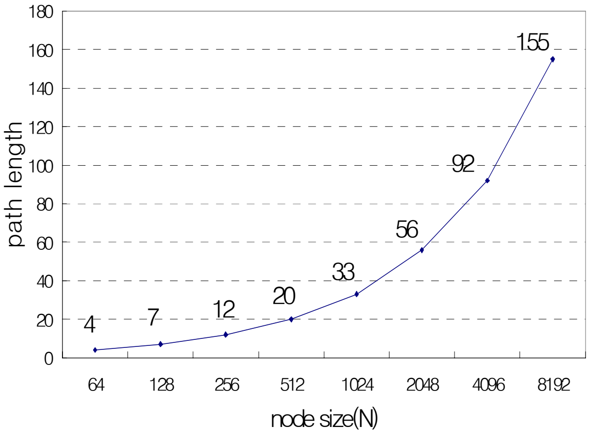


Fig 11 Path length vs. network size(N)

In figure 11, it represents the path length versus mobile ad-hoc network size ($N = 2^m, m = 6, 7, ..., 13)$. As shown in the figure, some path lengths are appeared to be smaller than $\log_2 N$, but gradually increasing in case that is bigger than $N(=128)$. Thus, Our proposed protocol in this paper is required to trade-off between the size of node repository ($\log_2 N$) and the length of certification path as the size of mobile ad-hoc network is increasing. Although the length of certification path is increasing as N is increasing, it can be noticed that the path length versus $N(<8{,}192)$ is smaller than $2\sqrt{N}$ [11]. Another issue of simulation is the possibility of certification path construction due to mobility of the node. Namely, it is very critical factor to recognize the possibility of path construction for evaluating the performance of the proposed algorithm according to the reducing rate $r(=(N-N_r)/N)$ of nodes in mobile ad-hoc network.

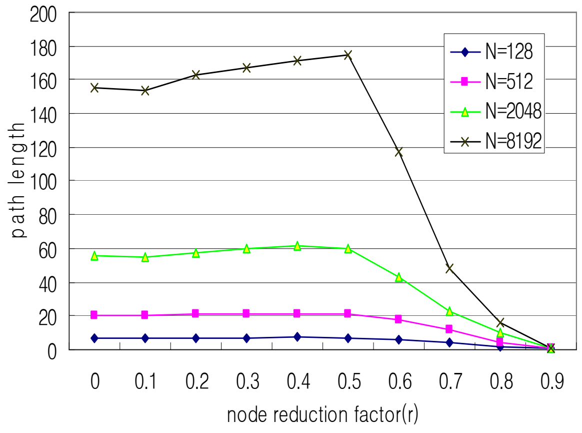


Fig 12 Path length vs. node reduction factor(r)

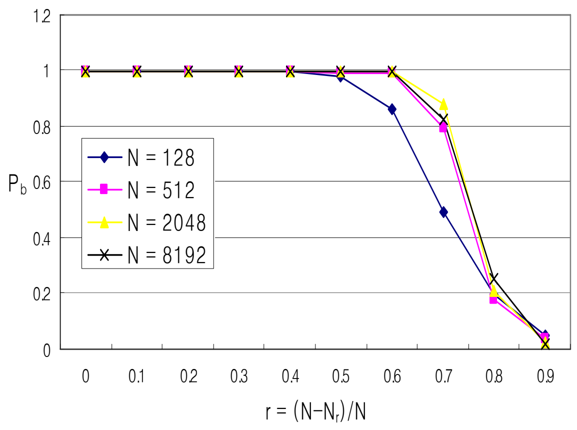


Fig 13 Success probability vs. node reduction factor

Figure 12 and 13 represent the path length and success probability of path construction versus node reduction factor $r(\in\{0.1, 0.2, ..., 0.9\})$ according to the network size $N(\in\{128, 512, 2{,}048, 8{,}192\})$. In Figure 12, it is shown that the path length is increased up to r (=0.4), but reduced from 0.5. Besides, as seen the success probability (P_b) of path construction in figure 13, it is shown that path construction attains about 97.6% success probability at confidence interval 99% in case of r = 0.5. In case that

r is 0.5, the real network size Nr is the same with $N/2(=2^{m-1})$. Thus, if r is bigger than 0.5, the network size N can be manipulated the new network size N' ($=N/2$) and the ID s of all nodes also will be manipulated smaller than N'. Through this approach, we can improve the performance as manipulating r to $(N'-Nr)/N'$. To change a node ID means updating public-key certificate so that a node ID lie on $0 \le node\ ID < N'$ range as explained in chapter 3, "Certificate Update".

6. Conclusion

We present the public-key management protocol and certification path construction algorithm improving the repository size belonging to each node as $\log_2 N$ in mobile ad-hoc network, introducing the concept of Hamming distance. For this method, each node is given unique ID and only issues, updates, and revokes public-key certificates of nodes that the Hamming distance against its own ID has "1". In this algorithm, each node uses the intermediate nodes to construct the certificate path from Tn to Vn. This can find a certification path by above 97.9% success probability, in case that r is smaller than 0.5 at confidence interval 99%. But depending on r, there is a disadvantage that certification path gets a little longer. Basic operations and certification path construction algorithm for public-key management are designed based on assumption that all nodes in mobile ad-hoc network are active. Therefore, a further study will be necessary for solving these problems.

References

[1] S. Corson and J. Macker, "Mobile Ad-hoc Networking (MANET): Routing Protocol Performance issues and Evaluation Considerations", IETF RFC2501, Jan. 1999.

[2] K. Fokine, "Key Management in Ad Hoc Networks", LiTH-ISY-EX-3322-2002, 2002.

[3] F. Stajano and R. Anderson, "The Resurrecting Duckling : Security Issues for Ad-hoc Wireless Networks", Proc. seventh Int'l workshop security protocols, 1999.

[4] H. Luo and S. Lu, "Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks", Technical Report 200030, UCLA Computer Science Department, Oct. 2000.

[5] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for MANET," Proc. ninth Int'l conf. Network Protocols (ICNP), Nov. 2001.

[6] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self-securing Ad Hoc Wireless Networks", Seventh IEEE Symp. on Computers and Communications (ISCC '02), 2002.

[7] A. Weimerskirch and G. Thonet, "A Distributed Light-Weight Authentication Model for Ad-hoc Networks", v. 2288 of LNCS, 2002.

[8] A. Herzberg, S. Jarecki, H. Krawczyk, M. Yung, "Proactive Secret Sharing, or: how to cope with perpetual leakage," Advances in Cryptography - Crypto 95' Proceedings, LNCS Vol 963, 1995.

[9] Y. Frankel, P. Gemmell, P.-D. MacKenzie, and M. Yung, "Optimal-Resilience Proactive Public-Key Cryptosystems", IEEE Symp. on Foundations of Computer Science, 1997.

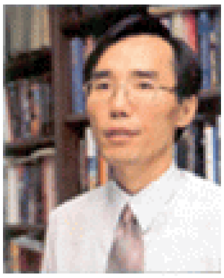
[10] J.-P. Hubaux, L. Buttyan, and S. Capkun, "The Quest for Security in Mobile Ad Hoc Networks," Proc. ACM Symp. Mobile Ad Hoc Networking and in Computing(MobiHoc), 2001.

[11] S. Capkun, L. Buttyan and J.-P. Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks", IEEE Trans. on mobile computing, vol. 2, No. 1, Jan./Mar. 2003.

[12] R. Hamming. Coding and Information Theory, Prentice-Hall, 1980.

[13] R. Housley, W. Polk, W. Ford, and D. Solo," Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", IETF RFC3280, April 2002.

Seok-Lae Lee received the B.S. and M.S. degrees in Electronic Communication Engineering from Hanyang University, Korea in 1992 and 1994. He is currently director of Korea Information Security Agency.



Joo-Seok Song received the B.S. degree in Electrical Engineering from Seoul National University, Seoul, Korea, in 1976, and the M.S. degree Electrical Engineering from KAIST, Korea, in 1979, and the Ph.D degree in Computer Science from University of California at Berkeley, U.S.A., in 1988. He is currently a Professor of Computer Science at Yonsei University, Seoul, Korea.